|  |
| --- |
| BUGT |
| Particle system research |

|  |
| --- |
| bas van rijn  4/3/2015 |

Particle system research

Ik heb de volgende IDE’s gebruikt aangezien ik die binnen bereik had:

**Unity3D 4.5.1f**

**Unreal 4 using unreal engine 4.7**

Ik wilde onderzoeken hoe beide IDE’s particles gebruiken en hoe je ze kan bewerken. Ik keek vooral naar hoe eenvoudig het kan aangezien wij in onze editor het onzelf ook makkelijk zullen moeten maken. Ik heb ook geprobeerd om particle systems die wij willen gebruiken al te bekijken (elementen zoals vuur water etc).

Tijdens mijn onderzoek kwam ik in de unity asset store een asset tegen met elemental particle systems. Uiteraard heb ik die meteen bekeken en ik denk dat het er in ieder geval mooi genoeg uitziet om als referentie te gebruiken voor onze game(kijken hoe zij het doen, kijken wat voor soort textures ze gebruiken etc). Ik het een build gemaakt en op de GIT gezet.

# Inhoud

[Unreal: 3](#_Toc413283424)

[Unity: 4](#_Toc413283425)

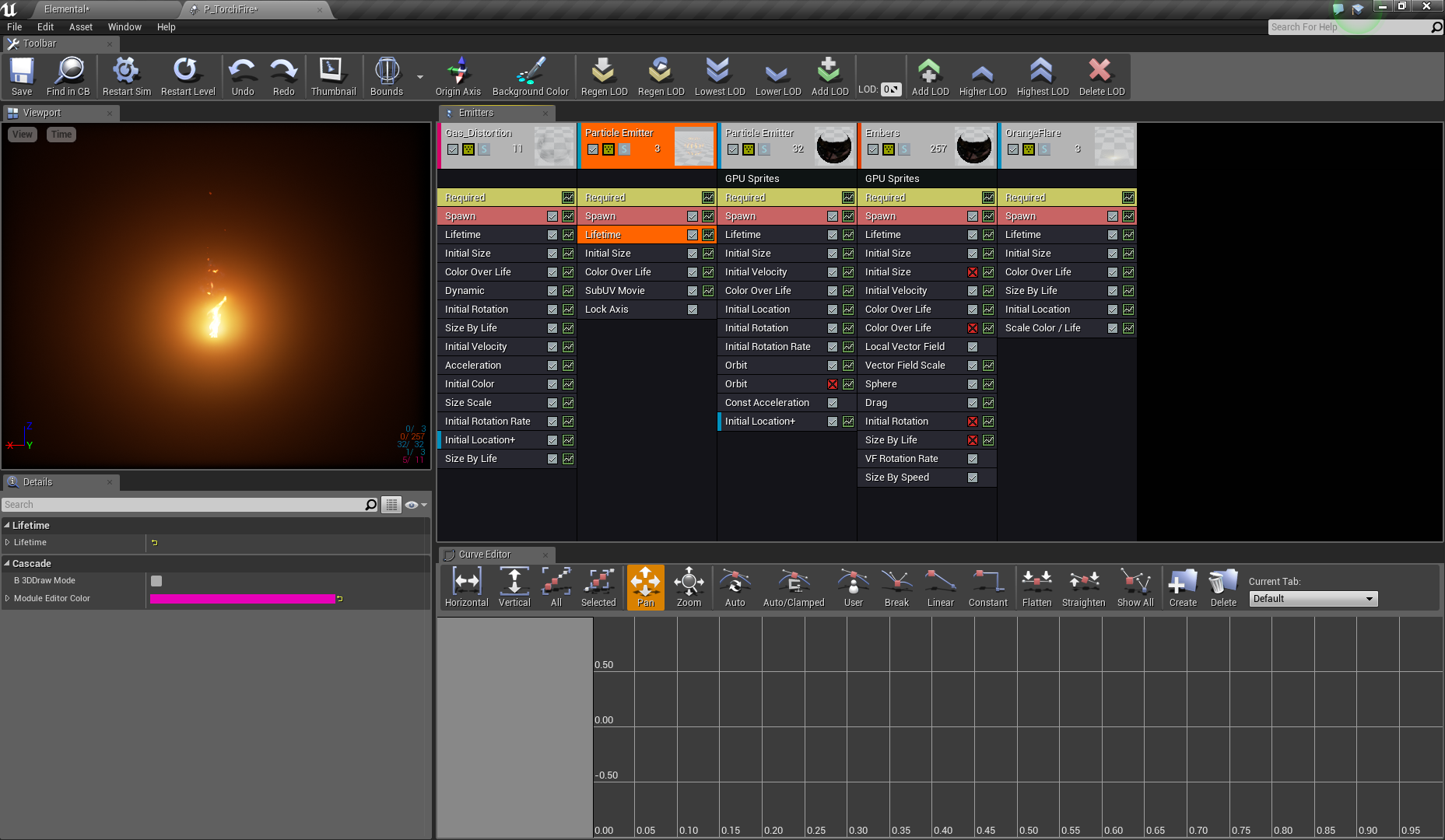
[Curves: 5](#_Toc413283426)

[Basis benodigdheden voor onze game: 6](#_Toc413283427)

[Advanced: 6](#_Toc413283428)

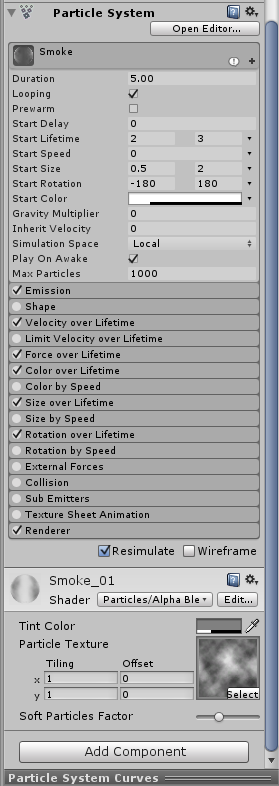
[Layout: 7](#_Toc413283429)

# Unreal:



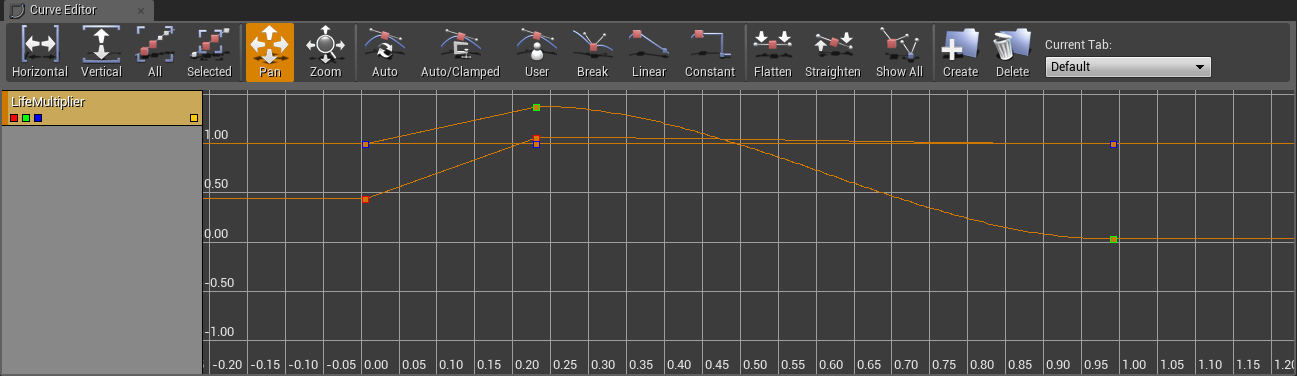
Dit is de particle system editor van Unreal. Volgens het irrlicht boek heet deze editor ‘cascade’. Deze editor vult het hele scherm en past de particle system direct aan, je ziet dus meteen het effect van je veranderingen. De particle system bestaat uit meerdere particle emitters. Deze emitters werken samen om het geheel te vormen. Elke emitter heeft weer meerdere modules. Elke module heeft een functie die de particles aanpast, zoals bijv; size by life, initial life, spawn en een required waar alle basic functionaliteit staat beschreven. Behalve de required en de spawn module hoeft de emitter geen modules te hebben maar zonder dat gebeurt er niks.

# Unity:

In Unity gebruiken ze niet het hele scherm maar gebruiken ze het tabje waar alle objecten mee geëdit worden. Het is een stuk minder uitgebreid en een stuk minder overzichtelijk. De particle system wordt ook meteen aangepast. Een particle system in unity heeft maar een emitter en als men meerdere effecten tegelijkertijd wil hebben(rook die van vuur af komt) dan moeten er meerdere particle sytem ‘objects’ gemaakt worden. Een particle sytem heeft meerdere modules die aangevinkt kunnen worden als ze gebruikt willen worden.

# Curves:

Ik vond in beide IDE’s dat particle system werken met curves. Een particle veranderd tijdens zijn lifetime(hij wordt kleiner, hij veranderd van kleur/doorzichtigheid), dit gebeurd vaak met een curve. Nu weet ik niet of wij dit kunnen implementeren maar het zou het wel een stuk makkelijker maken om mooie particles te maken. Sommige modules kunnen met een eenvoudige constante gemaakt worden(denk aan ‘particle.size -=size\*deltatime’ ofzoiets) maar er zijn ook momenten waar een curve toch echt mooier en makkelijker is.



Voorbeeld van de unreal IDE. Heel uitgebreid maar wel heel powerful en makkelijk te gebruiken.

# Basis benodigdheden voor onze game:

Ik heb mij hierbij vooral georienteerd op wat de Unity engine elemental demo gebruikt. Welke modules die particle systems gebruiken.

Initial(met welke waarde begint een particle):

* Size(min,max)
* Color
* Velocity(min,max)
* Loop
* Max particles
* Play on awake

Modules:

* Emission(constant of burst, rate)
* Shape(point, cube, sphere, cone, etc)
* Force over lifetime
* Color over lifetime
* Velocity over lifetime
* Rotation over lifetime

## Advanced:

Deze benodigdheden zijn niet noodzakelijk maar sommige particle systems zoals rain gebruiken ze wel en dus kunnen ze handig zijn te gebruiken.

Modules:

* Collision(Dit kan nog wel eens een lastige worden)
* Sub Emitter(na een collision wil je misschien nog iets emitten)
* Size by speed
* Rotation by speed

# Layout:

Met betrekking tot layout heb ik niet veel kunnen onderzoeken. Ik vond beide IDE’s goed genoeg werken(voor e.v.t. onze game). De manier waarop de systems werken in Unreal is wel handiger, meerdere emitters in een particle system. Ook vond ik het handiger dat als je een module niet gebruikt je hem niet ziet, in unity moet je de modules aanvinken als je ze gebruikt maar dat is wel iets onoverzichtelijker. Ook is het gebruik van sliders voor waardes veel gebruiksvriendelijker.